Global - Simple reg.
Global - Polynomial regression → $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \dots + \beta_d x_i^d + \varepsilon_i$
Global - step functions → K cutpoints (K+1 dummy variables)
Local - Piecewise polynomials
Local - Binning
Local - Local Averages
Local - KNN
Local = weighted local averaging
Local - = kernel estimation
Local - Local polynomials
Global - Splines < regression / smoothing

$$= \frac{P\_18\_boot - P\_18}{\sqrt{\frac{P\_18\_boot\,(1 - P\_18\_boot)}{n}}}$$

Simple regression

(Global) Polynomial Regression:
we add degrees
$y_i = \beta_0 + \sum_{j=1}^{d} \beta_j x_i^j + \varepsilon_i$

Step function:
we select **K** cutpoints;
$y_i = \beta_0 + \sum_{k=1}^{K} \beta_k C_k(x_i) + \varepsilon_i$
β_j = mean of the responses in $[c_j, c_{j+1})$

Piecewise Polynomial regression:
K cutpoints, in each region we have a polynomial (different degrees are ok)

Binning:
we divide the support in bins. Each bin's value is the average of the obs. inside the bin

Local averages:
we have a sliding window (fixed β), the value (of the regression) for the central point is the average of the points in the window

KNN
instead of a fixed window, for each point the value for the regr. is the mean of the closest **K** points

Weighted local average/ Kernel estimation:
greater weights on observations close to the focal point
(we weight points through the kernel of a density)

Local polynomials
extension of the kernel estimation to a polynomial fit of the focal point

SPLINES
- Regression:
we select a suitable basis functions that allows to fit in each bin a polynomial
(cubic splines/natural sp.)
- Smoothing:
it minimizes a penalization
(square error + λ (penalization))
Natural cubic splines are the best,

Full-conformal

$X_1, \dots, X_n \rightarrow X_{n+1}$?

- We create a grid of possible values for X_{n+1}
- for each point of the grid:
  - $\mathbb{X} = (X_1, \dots, X_n, X_{n+1})$ augmented sample

EXPENSIVE

  - for every X_j in $\mathbb{X}$ we calculate how much X_j is **weird** w.r.t. $\boxed{\mathbb{X}_{-j}}$ (:= score (the higher, the weirder the point))

changes for every j and for every point in the grid

  - p-value for the specific X_{n+1}:
    $$= \frac{\#(\text{scores} \geq \text{score}(X_{n+1}))}{n+1}$$
- The prediction interval is the set of points of the grid for which the p-value is $\geq$ treshold (α)

Split-conformal

$X_1, \dots, X_n \rightarrow X_{n+1}$?

- We split the data in
  $\mathbb{X}_1 = (X_1, \dots, X_m)$
  $\mathbb{X}_2 = (X_{m+1}, \dots, X_n)$
- we create a grid of possible values for X_{n+1}
- for each point of the grid:
  - $\mathbb{X}_2^* = (X_{m+1}, \dots, X_n, X_{n+1})$ augmented sample
  - for every X_j in $\mathbb{X}_2^*$ we calculate how much X_j is weird w.r.t. $\boxed{\mathbb{X}_1}$ (:= score (the higher, the weirder is the point))

stays always the same

  - p-value for the specific X_{n+1}:
    $$= \frac{\#(\text{scores} \geq \text{score}(X_{n+1}))}{\dim(\mathbb{X}_2^*)}$$
- The prediction interval is the set of points of the grid for which the p-value is $\geq \alpha$